

Softwareparadigmen SS 2015, Übungsblatt 2

Abgabe: 20. Mai 2015, bis 16:00 Uhr vor dem Sekretariat IST, Infeldgasse 16b, 2. OG

Beispiel 1 (1,5 P.)

Schreiben Sie eine Funktion in EXP über Listen und natürliche Zahlen inkl. Null $\mathbb{L} + \mathbb{N}_0$ (\mathbb{L} : first, rest, build, atom?, eq?, \mathbb{N}_0 : plus, minus, lt?, eq?, wobei jede Funktion auf \mathbb{N}_0 im Fehlerfall 0 zurückliefert), welches eine nicht-verschachtelte Liste entgegen nimmt und die Anzahl der in der Liste enthaltenen Zahlen retourniert.

Zeigen Sie die Korrektheit Ihrer Implementierung unter Verwendung der Interpretationsfunktion.

Beispiel 2 (3 P.)

Schreiben Sie ein Programm δgcd in EXP über den Datentyp der natürlichen Zahlen inkl. Null \mathbb{N}_0 (eq?, gt?, lt?, minus, plus), welches zwei Parameter entgegen nimmt und den größten gemeinsamen Teiler daraus berechnet.

Zeigen Sie die Korrektheit Ihrer Implementierung unter Verwendung der Interpretationsfunktion.

Hinweis: Die Verwendung des Modulo-Operators ist nicht erlaubt!

Beispiel 3 (3 P.)

Schreiben Sie ein Programm in EXP über den Datentyp der Listen und Binärziffern $\mathbb{L} + \mathbb{B}$ (\mathbb{L} : first, rest, build, atom?, eq?, \mathbb{B} : is0?, is1?), welches als Parameter zwei nicht-verschachtelte Listen (x, y) nimmt und eine elementweise mit Bijunktion verknüpfte Liste zurückgibt.

Die Bijunktion ist für zwei Binärziffern a, b definiert als

$$a \Leftrightarrow b = \begin{cases} 1 & \text{wenn } a = b \\ 0 & \text{sonst} \end{cases}$$

Beispielsweise würde ein Aufruf mit den Listen [1100] und [1010] das Ergebnis [1001] produzieren.

Zeigen Sie die Korrektheit ihrer Implementation für alle nicht-verschachtelten Listen von Binärziffern unter Verwendung der Interpretationsfunktion.

Beispiel 4 (1,5 P.)

- Schreiben Sie eine Scala Funktion **reverse**, die die Elemente einer übergebenen Liste von Integern in umgekehrter Reihenfolge zurückgibt. Das Verwenden der eingebauten Funktionen ist dabei nicht erlaubt.
- Beweisen Sie, dass gilt:
`reverse(append(as, bs)) = append(reverse(bs), reverse(as))`

Beispiel 5 (1,5 P.)

- Schreiben Sie eine Scala Funktion **without**, die *true* zurück gibt, wenn eine Ganzzahl in einer Liste **nicht** enthalten ist, andernfalls soll *false* retourniert werden.
- Zeigen Sie, dass gilt:
`without(x, append(as, bs)) = without(x, as) && without(x, bs)`

Beispiel 6 (2 P.)

Gegeben seien die beiden nachstehenden Funktionen **sum(as)** und **sum1(as, i)** die beide die Summe der Listenelemente berechnen. Zeigen Sie, dass gilt:
`sum(as) = sum1(as, 0)`.

```
def sum(as: List[Int]): Int = as match {  
  case Nil => 0  
  case x::xs => x + sum(xs)  
}
```

```
def sum1(as: List[Int], i: Int): Int = as match {  
  case Nil => i  
  case x::xs => sum1(xs, i + x)  
}
```

Viel Erfolg!